

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application. The following listing provides the amended claims with deleted material crossed out and new material underlined to show the changes made.

1. (Currently Amended) A method comprising:

for an event to be logged that has not yet been logged within an application:

creating an event object, said event object occupying a memory space that is independent of said application;

logging within said event object a start time, end time, and information regarding the event; and

analyzing ~~reviewing~~ at least one of said start time, end time, and information regarding the event,

wherein said creating, said logging, and said analyzing ~~reviewing~~ are performed by an event logging mechanism running independently from said application on a single computer on which said application executes.

2. (Previously Presented) A method according to claim 1 further comprising:

checking whether event logging has been turned on for the event.

3. (Previously Presented) A method according to claim 2, wherein said creating and said logging are performed for each event having event logging turned on, wherein a plurality of event objects are created and logged for a plurality of events.

4. (Previously Presented) A method according to claim 3, wherein said reviewing comprises analyzing said event object after event logging is turned off.

5. (Previously Presented) A method according to claim 4, wherein analyzing includes:

allowing user definition of the hierarchical levels of granularity of said events whose event objects are to be analyzed; and

allowing user definition of contexts for differentiating repeated occurrences of events deemed identical by nature of their hierarchical position.

6. (Previously Presented) A method according to claim 5, wherein analyzing further includes:

grouping events into their hierarchical subgroups; and

grouping events by their context, if any are defined.

7. (Previously Presented) A method according to claim 6, wherein analyzing comprises:

traversing through the hierarchy of subgroups until the subgroup of finest granularity is traversed;

subdividing said events into further subgroups;

computing statistics for each subgroup while traversing; and

displaying said statistics.

8. (Previously Presented) A method according to claim 7, wherein if said subgroup of finest granularity has been traversed, then:

aggregating events deemed identical by virtue of their hierarchical position into an aggregate;

computing statistics for each aggregate; and

displaying said statistics for each said aggregate.

9. (Previously Presented) A method according to claim 7, wherein said analyzing includes:

aggregating events deemed identical by virtue of their context into an aggregate;

computing statistics for each aggregate; and
displaying said statistics for each said aggregate.

10. (Previously Presented) A computer comprising computer readable storage for storing:

a foundational layer upon which applications are built or executed; and
an event logging mechanism created by said foundational layer, said logging mechanism executing independently of said applications, said mechanism for:

identifying a set of events for an application executing on said foundational layer,

generating an event log for the application, and

analyzing the event log, said event log generated without referencing any event logs of said application, wherein each of said events is designated an enabled/disabled status, wherein a disabled status disables all logging for an event, wherein said event logging mechanism performs the identifying, generating, and analyzing on said computer on which said application executes.

11. (Previously Presented) A computer according to claim 10, wherein said event logging mechanism logs start time, end time, and other event information into an event object for each event to be logged.

12. (Previously Presented) A computer according to claim 10, wherein said foundational layer is an operating system.

13. (Previously Presented) A computer according to claim 10, wherein said foundational layer is a programmable framework.

14. (Previously Presented) A computer according to claim 10, wherein said event logging mechanism can be turned on and then off from beyond an execution space of said applications within said foundational layer, said turning on and off separate for each event.

15. (Previously Presented) A computer according to claim 10, wherein said event logging mechanism can be turned on and turned off and configured using a web browser application.

16. (Previously Presented) A computer according to claim 15, wherein said event logging mechanism generates a plurality of event objects and is configured to analyze said event objects and present to said browser application the results thereof.

17. (Previously Presented) A computer according to claim 16, wherein said event logging mechanism is configured to analyze said event objects based upon hierarchical and contextual grouping.

18. (Previously Presented) A computer to claim 16, wherein said event logging mechanism is configured to aggregate said event objects deemed identical based upon at least one of hierarchical and contextual grouping.

19. (Currently Amended) An article comprising a computer readable medium storing a computer program for execution by at least one processor, the computer program comprising a set of instructions which when executed causes:

for each event in a plurality of events to be logged that has not yet been logged within an application,

creating an event object, said event object occupying a memory space that is independent of said application;

logging within said event object a start time, end time, and information regarding the event; and

analyzing at least one of said start time, end time, and information regarding the event, wherein said creating, logging, analyzing are performed by an event logging mechanism running independently from said application on a single computer on which said application executes.

20. (Previously Presented) An article according to claim 19, wherein the computer program further comprises a set of instructions which when executed causes:

analyzing of said event objects according to hierarchical and contextual grouping.

21. (Canceled)

22. (Canceled)

23. (Previously Presented) A computer comprising computer readable storage for storing:

a foundational layer upon which applications are executed;

an event-logging mechanism for execution on said foundational layer, said mechanism executing independently of said applications, said mechanism for:

identifying a set of events for an application executing on said foundational layer, and

generating a hierarchical event log for display in a web browser, said event log generated without referencing any event logs of said application and comprising an event in the hierarchy that includes a sub-event, wherein said event-logging mechanism performs the identifying and generating on said computer on which said application executes.

24. (Previously Presented) A computer according to claim 23, wherein said generating an event log comprises storing, for each event to be logged, a temporal attribute of an event in an event object associated with the event.

25. (Previously Presented) A computer according to claim 23, wherein said event-logging mechanism is further for analyzing said event log according to hierarchical and contextual grouping.

26. (Previously Presented) A computer according to claim 23 further comprising a first area of memory allocated to the application, a second area of memory allocated to the event logging mechanism, wherein said first area of memory allocated to the application is separate from the second area allocated to the event logging mechanism.

27. (Previously Presented) A computer according to claim 23, wherein said event logging mechanism can be turned on, turned off, and configured using the web browser.

28. (Previously Presented) A computer according to claim 23, wherein said event logging mechanism is further for allowing a user to enable and disable event logging for each event in the set of events, wherein said generating the event log is performed for each event having event logging enabled.

29. (Previously Presented) A computer according to claim 23, wherein the foundational layer is an operating system upon which applications are executed.

30. (Currently Amended) An event logging method comprising:
for each of a plurality of events that need to be logged but have not yet been logged within a plurality of applications:

creating an event object;

storing said event object in a first memory space that is uniquely allocated for the event logging method, said first memory space separate from a second memory space allocated for the plurality of applications;

logging within said event object a start time, end time, and information regarding the event; and

analyzing at least one of said start time, end time, and information regarding the event, wherein said creating, storing, logging, and analyzing are performed by an event logging mechanism running independently from said applications on a [[a]] single computer.

31. (Previously Presented) A method according to claim 30 further comprising creating, for the event, an enabled/disabled status wherein the disabled status disables all logging for the event within a system that includes the plurality of applications.

32. (Previously Presented) A method according to claim 30 further comprising checking, for each event identified by an application within the plurality of applications, whether event logging has been enabled.

33. (Previously Presented) A method according to claim 32, wherein said analysis is performed after event logging is disabled.

34. (Previously Presented) A method according to claim 30, wherein the memory space occupied by the event log is within memory space that has been allocated solely to the event logging mechanism.

35. (Previously Presented) A method according to claim 30, wherein the events that are logged by the event logging mechanism have not been previously logged by any other application.

36. (Previously Presented) A method according to claim 30, wherein information placed in the event log is first logged by the event logging mechanism.

37. (Previously Presented) A method according to claim 30 further comprising an enable/disable state for each event, wherein the disable state precludes any system from creating an event log.

38. (Previously Presented) A method according to claim 30, wherein said creating is done by a foundational layer that is a development framework.

39. (Currently Amended) A method of logging events for an application, said method comprising:

identifying a set of events generated by said application; and

hierarchically logging said identified set of events, wherein at least one event in the hierarchy comprises a sub-event, wherein said identifying and said logging are performed by an event logging mechanism running independently from said application on a single computer on which said application executes.

40. (Previously Presented) A method according to claim 39 further comprising:

checking whether event logging has been turned on for the event;

wherein said creating and said logging are performed for each event having event logging turned on, wherein a plurality of event objects are created and logged for a plurality of events.

41. (Previously Presented) A method according to claim 40 further comprising:

analyzing said event objects after event logging is turned off.

42. (Previously Presented) A computer comprising storage for:

a foundational layer upon which applications are executed; and

an event-logging mechanism for execution on said foundational layer, for functioning interoperably with but separately from said applications, said mechanism for:

identifying a set of event data for an application executing on said foundational layer,

generating an event log to record said identified event data,

analyzing said event data, wherein said application does not generate an event log.

43. (Previously Presented) A computer according to claim 42, wherein said generating an event log comprises storing, for each event to be logged, a start time, end time, and information regarding the event.

44. (Previously Presented) A computer according to claim 42, wherein said event-logging mechanism comprises analyzing of said event log according to hierarchical and contextual grouping.

45. (Previously Presented) A computer according to claim 44 further comprising an enable/disable state for each event identified by the application, wherein the disable state precludes the system from creating an event log, wherein generating an event log is performed for each event having event logging enabled.

46. (Currently Amended) A method comprising:

- identifying a set of events that has not been logged by ~~for~~ an application;
- analyzing the set of events identified for the application, wherein each event comprises at least a start time and an end time;
- grouping the set of events based on said analysis of the set of events; and
- generating a display of said set of events based on said grouping, wherein said identifying, analyzing, grouping, and generating are performed by an event analysis mechanism running on a single computer on which said application runs, said event analysis mechanism running independently from said application on said computer.

47. (Previously Presented) The method of claim 46, wherein said set of events are identified by an event logging mechanism prior to performing said analyzing, grouping, and generating.

48. (Previously Presented) The method of claim 46, wherein said generated display is for displaying said groupings of said set of events in a web browser.

49. (Previously Presented) The method of claim 46, wherein said grouping comprises grouping said set of events into a hierarchy, wherein at least one event in the hierarchy includes a sub-event.

50. (Previously Presented) The method of claim 46 further comprising computing statistics using the start time and the end time of the set of events, wherein said generating comprises including the statistics in the display.